

1 CLAIMS

2
3 1. A method for processing a database query according to at least one
4 grouping column value, the method comprising:

5 partially pre-aggregating records in a database to provide a result that
6 contains at least two records having like grouping column values; and

7 aggregating records derived from the partial pre-aggregation to provide a
8 result that contains records having unique grouping column values.
9

10 2. The method as recited in claim 1, wherein the partially pre-
11 aggregating further comprises:

12 maintaining a record store in memory, the record store having one record
13 for each different grouping column value encountered in the operation;

14 receiving a new record;

15 combining the new record with a record having the same grouping column
16 value, if such a record exists; and

17 adding the new record to the record store in the memory if there is no
18 record in the record store that has the same grouping column value as the new
19 record.
20

21 3. The method as recited in claim 2, further comprising:

22 adding additional new records to the record store until the record store
23 reaches a capacity such that it can accept no new records; and

24 outputting one or more records from the record store to a subsequent
25 database operator.

1
2 4. The method as recited in claim 3, wherein after the one or more
3 records have been output to the subsequent database operator, the adding and
4 outputting are repeated until there are no new records to process.

5
6 5. The method as recited in claim 4, wherein any records remaining in
7 the record store after there are no new records to process are output to the
8 subsequent database operator.

9
10 6. The method as recited in claim 3, wherein the subsequent database
11 operator is a join.

12
13 7. The method as recited in claim 1, further comprising estimating the
14 costs and benefits of the partial pre-aggregation, and partially pre-aggregating the
15 records only if the estimating indicates that the benefits are greater than the costs.

16
17 8. The method as recited in claim 1, wherein the partially pre-
18 aggregating includes utilizing a hashing function.

19
20 9. The method as recited in claim 1, wherein the partial pre-aggregating
21 creates a record store in memory, and wherein the method further comprises
22 utilizing the record store in memory for one or more other database operators.

1 10. One or more computer-readable media having computer-executable
2 instructions that, when executed by a computer, perform the method recited in
3 claim 1.

4
5 11. A computer programmed to perform the method recited in claim 1.

6
7 12. A relational database system, comprising:
8 memory for storing a record store, the memory having a portion available
9 for query processing;
10 a query processor coupled to the memory to process a query on the record
11 store according to at least one grouping column value, the query processor being
12 configured to partially pre-aggregate the record store to provide a result that
13 contains at least two data records that have like grouping column values; and
14 the query processor being further configured to aggregate data records
15 resulting from the partial pre-aggregation to provide an aggregation result that
16 contains data records, no two of the data records having a same grouping column
17 value.
18
19
20
21
22
23
24
25

1
2 **13.** The relational database system as recited in claim 12, wherein the
3 query processor being configured to partially pre-aggregate the record store
4 comprises the query processor being configured to:

5 maintain a record store in the volatile memory, the record store having one
6 record for each different grouping column value encountered in the partial pre-
7 aggregation;

8 receive an input record from the non-volatile memory;

9 combine the input record with a record in the record store that has the same
10 grouping column value, if there is such a record; and

11 adding the input record to the record store if there is no record in the record
12 store that has the same grouping column value as the input record.
13

14 **14.** The relational database system as recited in claim 13, wherein the
15 query processor is further configured to:

16 add additional input records to the record store in memory until the record
17 store reaches a capacity such that it can accept no more input records; and

18 output the records in the record store to a subsequent database operator.
19

20 **15.** The relational database system as recited in claim 14, wherein the
21 query processor is configured to:

22 continually add input records to the record store; and

23 output one or more records from the record store to a subsequent database
24 operator when the record store reaches a capacity such that it can accept no more
25 new records, or whenever there are no new records to process.

1
2 **16.** The relational database system as recited in claim 12, wherein the
3 query processor is further configured to perform a join on the records resulting
4 from the partial pre-aggregation.
5

6 **17.** The relational database system as recited in claim 12, wherein the
7 query processor is further configured to create a record store as a result of the
8 partial pre-aggregation and utilize the record store in processing of another
9 database operator.
10

11 **18.** The relational database system as recited in claim 12, further
12 comprising:
13

14 a query optimizer configured to estimate the costs and benefits of the query
15 processor performing a partial pre-aggregation; and
16

17 wherein the query processor performs the partial pre-aggregation only if the
18 query optimizer indicates that the benefits of the partial pre-aggregation are
19 greater than the costs of the partial pre-aggregation.
20

21 **19.** The relational database system as recited in claim 12, wherein the
22 query processor is further configured to utilize hashing to perform the partial pre-
23 aggregation.
24
25

1 **20.** The relational database system as recited in claim 12, wherein the
2 query processor is further configured to utilize hashing and partitioning to perform
3 the partial pre-aggregation.

4
5 **21.** A relational database computer program embodied on a computer-
6 readable medium, comprising:

7 partial pre-aggregation code to partially pre-aggregate data records
8 according to at least one grouping column value to provide a partial pre-
9 aggregation result having two or more records having like grouping column
10 values; and

11 aggregation code to aggregate data records in the partial pre-aggregation
12 result to provide an aggregation result having records with unique grouping
13 column values.
14
15
16
17
18
19
20
21
22
23
24
25

1
2 **22.** The relational database computer program as recited in claim 21,
3 wherein the partial pre-aggregation code is designed to:

4 maintain a record store in memory, the record store having one record for
5 each different grouping column value encountered in the partial pre-aggregation;

6 receive a new record;

7 combine the new record with a record in the record store having the same
8 grouping column value, if such a record exists;

9 add the new record to the record store no record in the record store that has
10 the same grouping column value as the new record;

11 continuously add additional new records to the record store until the record
12 store has reached a record store capacity;

13 output one or more records from the record store to a subsequent database
14 operator when the record store has reached the record store capacity; and

15 output the records in the record store when there are no new records to
16 process.

17
18 **23.** The relational database computer program as recited in claim 22,
19 further comprising database operator code that utilizes the record store for input.
20
21
22
23
24
25

1
2 **24.** A relational database computer program stored on a computer-
3 readable medium, the relational database computer program comprising computer-
4 executable instructions that, when executed on a computer, perform the following
5 steps:

6 receiving a stream of input records;
7 aggregating each input record in the stream as it is received to create a
8 record store;
9 joining records in the record store with other data; and
10 aggregating the records output from the join.

11
12 **25.** The relational database computer program as recited in claim 24,
13 wherein:

14 the record store has a capacity that is less than the number of records in the
15 stream of input records; and

16 the aggregating each input record is performed until the record store
17 reaches capacity.

18
19 **26.** The relational database computer program as recited in claim 24,
20 further comprising computer-executable instructions that, when executed by a
21 computer, perform the following steps:

22 determining if it is optimal to aggregate the input records prior to
23 performing the join; and

24 performing the aggregation prior to the join only if a determination is made
25 that it is optimal to perform an aggregation prior to the join.